NASA-CR-192948

NEURAL NETWORKS: WHAT NON-LINEARITY TO CHOOSE

Vladik Kreinovich, Chris Quintana*

Abstract. Neural networks are now one of the most successful learning formalisms. Neurons transform inputs $x_1, ..., x_n$ into an output $f(w_1x_1 + ... + w_nx_n)$, where f is a non-linear function and w_i are adjustable weights. What f to choose? Usually the logistic function is chosen, but sometimes the use of different functions improves the practical efficiency of the network.

We formulate the problem of choosing f as a mathematical optimization problem and solve it under different optimality criteria. As a result, we get a list of functions f that are optimal under these criteria. This list includes both the functions that were empirically proved to be the best for some problems, and some new functions that may be worth trying.

1. FORMULATION OF THE PROBLEM.

Neural networks are now one of the most successful learning formalisms (see, e.g., the recent survey in [Hecht-Nielsen 1991]). After the initial success of linear neural models, in which the output y is equal to the linear combination of the input signals x_i, i.e. $y = w_1x_1 + w_2x_2 + ...$, it was shown in [Minsky Papert 1968] that if we only have linear neurons, then we end up with only linear functions and this severely limits the number of problems that we can solve using the network. The next step, then, is to consider non-linear neurons, in which the output signal is equal to $f(w_1x_1 + w_2x_2 + ...)$, where $f(y)$ is a given non-linear function. A natural question arises: **what function $f(y)$ do we choose?**

Why is this problem important? It is a very important problem because although neural networks help us to design good learning procedures, these procedures are far from being reliable. Sometimes these procedures do not work; sometimes they work but demand too much time, and too big a sample, to learn. Naturally, we might think that this is because the function f that we used was not the best one. Sometimes the use of different functions can improve the practical efficiency of the network (see, e.g., [Wasserman 1989, pp. 15-16]). If a simple guess can really improve the learning performance, then it is natural to suppose that deep mathematical optimization will lead to even better results.

Why is this problem difficult? We want to find a function f for which some characteristics J of learning, such as average learning time or average number of errors, is optimal (in these cases minimal). The problem is that even for the commonly used logistic function (see below), we do not know how to compute any of these possible characteristics. How can we find f for which $J(f)$ is optimal if we cannot compute $J(f)$ even for a single f? There does not seem to be a likely answer.

However, we will show that this problem is solvable (and give the solution) using advanced math, namely, group theory. (For a general idea of this approach, see [Kreinovich 1990].)

2. WHAT IS KNOWN.

The first non-linear neuron was proposed in [Cowan 1967]. Cowan chose the logistic function, $s_0(y) = 1/(1 + \exp(-y))$, because it leads to a good approximation of the behavior of real (i.e. biological) neurons. The properties of neural networks with different f were studied by Grossberg (see, e.g., [Grossberg 1988]) who showed that the logistic function has several nice properties useful for learning and is therefore an adequate choice. His analysis restricted the class of possible functions, but there are still many other functions with the same properties. So we still have to make a choice. Another attack was undertaken by Hecht-Nielsen, who in [1987] added a demand that any function must be approximable by some neural network. This

* Computer Science Department, University of Texas at El Paso, El Paso, TX 79968, USA

approach leads to non-trivial mathematics, but the final result (see, e.g., [Kreinovich 1991]) is that any smooth function f will work, so this additional demand does not help us to choose f.

3. WHAT DO WE PROPOSE

3.1. Motivations of the proposed mathematical definitions.

We must choose a family of functions, not a single function. We speak about choosing f, but the expression for $f(y)$ will change if we change the units in which we measure all the signals (input, output and intermediate), so in mathematical terms, it is better to speak about choosing a family of functions f. It is reasonable to suggest that if an f belongs to this family, then this family must contain kf for positive real numbers k. This corresponds to changing units. Also, it must contain $f + c$, where c is a constant. This is equivalent to adding a constant bias and therefore does not change the abilities of the resulting network. Since we are talking about non-linear phenomena, we can also assume that some non-linear "rescaling" transformations $x \to g(x)$ are also applicable, i.e., the family must include the composition $g(f(y))$ for each of functions f. This family must not be too big, therefore, it must be determined by finitely many parameters and should ideally be obtained from one function $f(y)$ by applying all these transformations. Without loss of generality, we can assume that this set of transformations is closed under composition and under inverse, i.e., if $z \to g_1(z)$ and $z \to g_2(z)$ are possible transformations, then $z \to g_1(g_2(z))$ and $z \to g_1^{-1}(z)$ are possible transformations, where by g_1^{-1} we denoted an inverse function $g_1^{-1}(z) = w$ if and only if $g_1(w) = z$. In mathematical terms this means that these transformations form a *group*, and therefore *a family is obtained by applying to some function $f(y)$ all transformations from some finite-dimensional transformation group G that includes all linear transformations (and maybe some non-linear ones).*

All these transformations correspond to appropriate "rescalings". Rescaling is something that is smoothly changing the initial scale. This means that if we have two different transformations, there must be a smooth transition between them. In mathematical terms, the existence of this continuous transition is expressed by saying that the group is *connected*, and the fact that both the transformations and the transitions are smooth is expressed by saying that this is a *Lie group* (see, e.g., Chevalley, 1946).

What family is the best? Among all such families, we want to choose the best one. In formalizing what "the best" means we follow the general idea outlined in [Kreinovich 1990]. The criteria to choose may be computational simplicity, efficiency of training, or something else. In mathematical optimization problems, numeric criteria are most frequently used, when to every family we assign some value expressing its performance, and choose a family for which this value is maximal. However, it is not necessary to restrict ourselves to such numeric criteria only. For example, if we have several different families that have the same training ability A, we can choose between them the one that has the minimal computational complexity C. In this case, the actual criterion that we use to compare two families is not numeric, but more complicated: *a family F_1 is better than the family F_2 if and only if either $A(F_1) > A(F_2)$ or $A(F_1) = A(F_2)$ and $C(F_1) < C(F_2)$.* A criterion can be even more complicated. What a criterion *must* do is to allow us for every pair of families to tell whether the first family is better with respect to this criterion (we'll denote it by $F_1 > F_2$), or the second is better ($F_1 < F_2$) or these families have the same quality in the sense of this criterion (we'll denote it by $F_1 \sim F_2$). Of course, it is necessary to demand that these choices be consistent, e.g., if $F_1 > F_2$ and $F_2 > F_3$ then $F_1 > F_3$.

Another natural demand is that this criterion must choose a *unique* optimal family (i.e., a family that is better with respect to this criterion than any other family). The reason for this demand is very simple. If a criterion does not choose any family at all, then it is of no use. If several different families are "the best" according to this criterion, then we still have a problem to choose among those "best". Therefore, we need some additional criterion for that choice. For example, if several families turn out to have the same training ability, we can choose among them a family with minimal computational complexity. So what we actually do in this case is abandon that criterion for which there were several "best" families, and consider a new

"composite" criterion instead: F_1 is better than F_2 according to this new criterion if either it was better according to the old criterion or according to the old criterion they had the same quality and F_1 is better than F_2 according to the additional criterion. In other words, if a criterion does not allow us to choose a unique best family it means that this criterion is not ultimate; we have to modify it until we come to a final criterion that will have that property.

The next natural condition that the criterion must satisfy is connected with the following. Suppose that instead of a neuron with the transformation function $f(y)$ we consider a neuron with a function $\bar{f}(y) = f(y+a)$, where a is a constant. This new neuron can be easily simulated by the old ones: namely, the output of this new neuron is $\bar{f}(w_1x_1 + w_2x_2 + ...) = f(w_1x_1 + w_2x_2 + ... + a)$, so it is equivalent to an old neuron with an additional constant input a. Likewise, the old neuron is equivalent to the new neuron with an additional constant input $-a$. Therefore, the networks that are formed by these new neurons have precisely the same abilities as those that are built from the old ones. We cannot claim that the new neurons have the same quality as the old ones, because adding a can increase computational complexity and thus slightly worsen the overall quality. But it is natural to demand that adding a does not change the *relative* quality of the neurons, i.e., if a family $\{f(y)\}$ is better that a family of $\{g(y)\}$, then for every a the family $\{f(y+a)\}$ must be still better than the family $\{g(y+a)\}$.

3.2. Definitions and the main result. By a *transformation* we mean a smooth (differentiable) function from real numbers into real numbers. By an *appropriate transformation group* G we mean a finite-dimensional connected Lie group of transformations. By a *family* of functions we mean the set of functions that is obtained from a smooth (everywhere defined) non-constant function $f(y)$ by applying all the transformations from some appropriate transformation group G. Let us denote the set of all the families by F.

A pair of relations $(<, \sim)$ is called *consistent* [Kreinovich 1990, Kreinovich Kumar 1990] if it satisfies the following conditions: (1) if $a < b$ and $b < c$ then $a < c$; (2) $a \sim a$; (3) if $a \sim b$ then $b \sim a$; (4) if $a \sim b$ and $b \sim c$ then $a \sim c$; (5) if $a < b$ and $b \sim c$ then $a < c$; (6) if $a \sim b$ and $b < c$ then $a < c$; (7) if $a < b$ then $b < a$ or $a \sim b$ are impossible.

Assume a set A is given. Its elements will be called *alternatives*. By an *optimality criterion* we mean a consistent pair $(<, \sim)$ of relations on the set A of all alternatives. If $a > b$, we say that a is *better* than b; if $a \sim b$, we say that the alternatives a and b are *equivalent* with respect to this criterion. We say that an alternative a is *optimal* (or *best*) with respect to a criterion $(<, \sim)$ if for every other alternative b either $a > b$ or $a \sim b$.

We say that a criterion is *final* if there exists an optimal alternative, and this optimal alternative is unique.

Comment. In the present section we consider optimality criteria on the set F of all families.

By the *result of adding* a to a function $f(y)$ we mean a function $\bar{f}(y) = f(y+a)$. By the *result of adding* a to a family F we mean the set of the functions that are obtained from $f \in F$ by adding a. This result will be denoted by $F + a$. We say that an optimality criterion on F is *shift-invariant* if for every two families F and G and for every number a, the following two conditions are true:

i) if F is better than G in the sense of this criterion (i.e., $F > G$), then $F + a > G + a$;

ii) if F is equivalent to G in the sense of this criterion (i.e., $F \sim G$), then $F + a \sim G + a$.

Comment. As we have already remarked, the demands that the optimality criterion is final and shift-invariant are quite reasonable. The only problem with them is that at first glance they may seem rather weak. However, they are not, as the following Theorem shows:

By a *logistic* function we mean $s_0(y) = 1/(1 + \exp(-y))$.

22-3

THEOREM 1. *If a family F is optimal in the sense of some optimality criterion that is final and shift-invariant, then every function f from F is equal either to $a + bs_0(Ky + l)$ for some a, b, K and l, or a linear function $a + bx$, or to $a + b\exp(Kx)$ for some a, b, K.*

Comments. 1. Logistic, hyperbolic-tangent, linear and exponential functions are really among the most popular [Kosko 1992].

2. We assumed that f must be smooth. If we consider f that can be not smooth in some points, then it is natural to assume that on the intervals on which f is smooth, it must coincide with one of these functions. Such piecewise smooth functions have also been successfully used, the most popular are *threshold* functions that are obtained from the smooth ones by restricting their values to $[0, \infty)$ or $[0, 1]$ [Kosko 1992].

(The proofs are given in Section 5).

4. OPTIMIZATION OF NEURAL NETWORKS: RELATED RESULTS

4.1, Scale-invariance instead of shift invariance. In the above text we assumed that the optimality criterion is shift-invariant. The same arguments can be used to motivate the demand that the optimality criterion is invariant with respect to *scaling transformations* $f(y) \to f(ay)$ for some $a > 0$. Let us analyze the consequences of this demand.

Definition. By the *result of rescaling* a function $f(y)$ by a real number $a > 0$ we mean a function $\bar{f}(y) = f(ay)$. By the *result of rescaling* a family F by a we mean the set of the functions, that are obtained from rescaling $f \in F$ by a. This result will be denoted by aF. We say that an optimality criterion on F is *scale-invariant* if for every two families F and G and for every number a the following two conditions are true:

i)' if F is better than G in the sense of this criterion (i.e., $F > G$), then $aF > aG$;

ii)' if F is equivalent to G in the sense of this criterion (i.e., $F \sim G$), then $aF \sim aG$.

THEOREM 2. *If a family F is optimal in the sense of some optimality criterion that is final and scale-invariant, then every function f from F is equal to $f(y) = (A + By^{-\alpha})/(C + Dy^{-\alpha})$ for some A, B, C, D and $\alpha > 0$.*

Comments. 1. In particular, for $A = 0$, $B = C = D = 1$ and $\alpha = 2$ we get the Cauchy function $f(y) = 1/(1 + y^2)$, that is used in neural networks (see, e.g., [Hecht-Nielsen 1991]. If $B = 0$ and α is an integer, $\alpha > 1$, we get ratio-polynomial signal functions that have also been successfully used [Kosko 1992]. For $\alpha = 1$, $B = 0$ and $A = C = D = 1$ this function equals to the expression $x/(1 + x)$, which was analyzed in [Munro 1986]. So this Theorem gives a list of possible optimal non- linear neurons that generalizes Cauchy and ratio-polynomial functions.

2. By comparing the results of Theorems 1 and 2 one can conclude that a scale-invariant criterion cannot be shift-invariant: indeed, in this case we could apply Theorem 2, so f must be described by the above expression. But these functions are different from the functions from Theorem 1, and so due to Theorem 1 this criterion is not shift-invariant.

But what if we still want our criterion to be both shift- and scale-invariant? For standard neurons with non-linearity of the type $y = f(w_1x_1 + ... + w_nx_n)$ it is impossible; in section 4.3 we'll show that it is possible for a more general type of neurons.

4.2. More general families of neurons. A natural way to define a finite-dimensional family of functions is to fix finitely many functions $f_i(y)$ and consider their arbitrary linear combinations $\sum_i C_i f_i(y)$.

Definition. Let's fix an integer m. By a *basis* we mean a set of m smooth functions $f_i(y)$, $i = 1, 2, ..., m$. By a *m-dimensional family* of functions we mean all functions of the type $f(y) = \sum_i C_i f_i(y)$ for some basis $f_i(y)$, where C_i are arbitrary constants. The set of all m-dimensional families will be denoted by F_m.

Our definitions of the optimality criterion, final criterion and shift-invariant criterion can be applied to these families.

THEOREM 3. *If an m-dimensional family F is optimal in the sense of some optimality criterion that is final and shift-invariant, then every function f from F is equal to a linear combination of the functions of the type $y^p \exp(\alpha y)\sin(\beta y + \phi)$, where p is a non-negative integer, α, β and ϕ are real numbers.*

Comment. In particular, for $p = 1, \alpha = \beta = 0$ we get linear functions; for $p = \beta = 0$ we get exponential functions; for $p = \alpha = \phi = 0$ we get a sine function, that has been successfully used [Braham 1989, Braham Hamblen 1990].

4.3. What if we demand scale-invariance and shift-invariance of the optimality criterion? Neurons with different non-linearity. In all the above cases we considered neurons that transform the input signals $x_1, ..., x_n$ into $f(w_1x_1 + ... + w_nx_n)$. This means that we added only one type of non-linearity: $y \to f(y)$ for some non-linear f. Let us consider neurons with the most general type of non-linearity $y_1, ..., y_p \to f(y_1, ..., y_p)$, where f is an arbitrary non-linear function of p variables. As in the above case, linear transformations are easy to implement, therefore we can consider neurons of the type

$$x_1, ..., x_n \to f(w_{11}x_1 + ... + w_{1n}x_n, w_{21}x_1 + ... + w_{2n}x_n, ..., w_{p1}x_1 + ... + w_{pn}x_n),$$

where w_{ij} are weights.

Definition. Let's fix integers m and p. By a *basis* we mean a set of m smooth functions $f_i(y_1, ..., y_p)$, $i = 1, 2, ..., m$. By a *m-dimensional family of functions of p variables*, or *m,p-family* for short, we mean a family that is formed by all functions of the type $f(y_1, ..., y_p) = \sum_i C_i f_i(y_1, ..., y_p)$ for some basis $f_i(y_1, ..., y_p)$, where C_i are arbitrary constants. The set of all m,p-dimensional families will be denoted by $F_{m,p}$.

Comment. Since it is easy to implement arbitrary linear transformations, it is reasonable to demand (like we did above) that the relative quality of the family does not change if we apply a shift or a rescaling to all its functions. So we arrive at the following definitions:

Definitions. Suppose that a vector $\vec{a} = (a_1, ..., a_p)$ is given. By the *result of adding* $\vec{a}$ to a function $f(y_1, ..., y_p)$ we mean a function $\tilde{f}(y_1, ..., y_p) = f(y_1 + a_1, ..., y_p + a_p)$. By the *result of adding* $\vec{a}$ to a family F we mean the set of the functions, that are obtained from $f \in F$ by adding $\vec{a}$. This result will be denoted by $F + \vec{a}$.

By the *result of rescaling* a function $f(y_1, ..., y_p)$ by a vector $\vec{a} = (a_1, ...)$ with $a_i > 0$ we mean a function $\tilde{f}(y_1, ..., y_p) = f(a_1y_1, ..., a_py_p)$. By the *result of rescaling* a family F by $\vec{a}$ we mean the set of the functions, that are obtained from rescaling $f \in F$ by $\vec{a}$. This result will be denoted by $\vec{a}F$.

Now we can apply the definitions of the optimality criterion, final criterion, shift- and scale-invariant criterion to m-dimensional families.

THEOREM 4. *If an m,p-dimensional family F is optimal in the sense of some optimality criterion that is final, shift- and scale-invariant, then every function f from F is equal to a polynomial of $y_1, ..., y_p$.*

Comment. So if we consider neurons with a more complicated non-linearity, we get the GMBH network (see [Hecht-Nielsen 1991, Section 5.6.1]), which historically is the first commercially successful non-linear neuron.

4.4. A related question: how to modify the weights during the training? This question is related to the following problem: usually during training weights are changed linearly (like $w \to w + c$ for some constant a). However, sometimes some weights become so big that the

22-5

output of the corresponding neurons is close to 1. These neurons are called *saturated* (see, e.g., [Wasserman 1989, pp. 90-91]. Saturation extends the training time, bringing the whole training process into a *paralysis*. To overcome paralysis, non-linear weight transformations are used: $w \to g(w)$ for some non-linear g. The main purpose of this transformation is to get rid of big values, i.e., to transform the whole set of real numbers $(-\infty, \infty)$ into some limited interval of values $[-\Delta, \Delta]$. But there are many functions with this property. So the natural question arises: *what g is the best to choose?*

The same arguments as in Sections 2 and 3 can be used to conclude that what we really need to choose is a *family* of functions, not just a function g, and that it is reasonable to assume that this family is obtained from some function $g(w)$ by applying all the transformations from some appropriate transformation group.

If a function g is better than a function $\bar{g}$, then it is reasonable to assume that it will still be better if we first make one more standard training step $w \to w + a$ and only then apply the non-linear transformation. These two consequent steps are equivalent to one transformation $w \to g(w+a)$. So the above demand means that if $g(w)$ is better than $\bar{g}(w)$, then $g_1(w) = g(w+a)$ must be better than $\bar{g}_1(w) = \bar{g}(w + a)$. In other words, the optimality criterion must be shift-invariant. Let's turn to formal definitions.

Definitions. Let us fix some real number $\Delta > 0$. We say that a function is *bounded* if its values always belong to $[-\Delta, \Delta]$. By a *family of weight transformations* we mean the set of functions that is obtained from a smooth (everywhere defined) non-constant bounded function $f(w)$ by applying all the transformations from some appropriate transformation group G. Let us denote the set of all such families by F_w. Let us consider optimality criteria on this set F_w.

THEOREM 5. *If a family F_w is optimal in the sense of some optimality criterion that is final and shift-invariant, then every bounded function g from F is equal to $a + bs_0(Ky + l)$ for some a, b, K and l.*

Comment. Such functions really proved to be the best in overcoming paralysis [Wasserman 1989, pp. 90–91].

5. PROOFS.

Proof of Theorem 1. The idea of this proof is as follows: first we prove that the appropriate transformation group consists of fractionally-linear functions (in part 1), then we prove that the optimal family is shift-invariant (in part 2), and from that in part 3 we conclude that any function f from F satisfies some functional equations, whose solutions are known.

1. By an *appropriate group* we meant a connected finite-dimensional Lie group of transformations of the set of real numbers R onto itself that contains all linear transformations. Norbert Wiener asked to classify such groups for an n-dimensional space with arbitrary n, and this classification was obtained in [Guillemin Sternberg 1964] and [Singer Sternberg 1965]. In our case (when $n = 1$) the only possible groups are the group of all linear transformations and the group of all fractionally-linear transformations $x \to (ax + b)/(cx + d)$. In both cases the group consists only of fractionally linear transformations (the simplified proof for the 1-dimensional case is given in [Kreinovich 1987]; for other applications of this result see [Kreinovich Kumar 1990, 1991], [Kreinovich Corbin 1991], [Kreinovich Quintana 1991]).

2. Let us now prove that the optimal family F_{opt} exists and is *shift-invariant* in the sense that $F_{opt} = F_{opt} + a$ for all real numbers a. Indeed, we assumed that the optimality criterion is final, therefore there exists a unique optimal family F_{opt}. Let's now prove that this optimal family is shift-invariant (this proof is practically the same as in [Kreinovich 1990]). The fact that F_{opt} is optimal means that for every other F, either $F_{opt} > F$ or $F_{opt} \sim F$. If $F_{opt} \sim F$ for some $F \neq F_{opt}$, then from the definition of the optimality criterion we can easily deduce that F is also

optimal, which contradicts the fact that there is only one optimal family. So for every F either $F_{opt} > F$ or $F_{opt} = F$.

Take an arbitrary a and let $F = F_{opt} + a$. If $F_{opt} > F = F_{opt} + a$, then from the invariance of the optimality criterion (condition ii) we conclude that $F_{opt} - a > F_{opt}$, and that conclusion contradicts the choice of F_{opt} as the optimal family. So $F_{opt} > F = F_{opt} + a$ is impossible, and therefore $F_{opt} = F = F_{opt} + a$, i.e., the optimal family is really shift- invariant.

3. Let us now deduce the actual form of the functions f from the optimal family. If $f(y)$ is such a function, then the result $f(y + a)$ of adding a to this function f belongs to $F + a$, and so, due to 2., it belongs to F. But all the functions from f can be obtained from each other by fractionally linear transformations, so $f(y + a) = (A + Bf(y))/(C + Df(y))$ for some A, B, C and D. So we arrive at a functional equation for f. Let us reduce this equation to a one with a known solution. For that purpose, let us use the fact that fractionally linear transformations are projective transformations of a line, and for such transformations the cross ratio is preserved [Aczel 1966, Section 2.3], i.e., if $g(y) = (A + Bf(y))/(C + Df(y))$, then

$$\frac{g(y_1) - g(y_3)}{g(y_2) - g(y_3)} \frac{g(y_2) - g(y_4)}{g(y_1) - g(y_4)} = \frac{f(y_1) - f(y_3)}{f(y_2) - f(y_3)} \frac{f(y_2) - f(y_4)}{f(y_1) - f(y_4)}$$

for all y_i. In our case this is true for $g(y) = f(y + a)$, therefore for all a the following equality is true:

$$\frac{f(y_1 + a) - f(y_3 + a)}{f(y_2 + a) - f(y_3 + a)} \frac{f(y_2 + a) - f(y_4 + a)}{f(y_1 + a) - f(y_4 + a)} = \frac{f(y_1) - f(y_3)}{f(y_2) - f(y_3)} \frac{f(y_2) - f(y_4)}{f(y_1) - f(y_4)}.$$

The most general continuous solutions of this functional equation are given by Theorem 2.3.2 from [Aczel 1966]: either f is fractionally linear, or $f(y) = (a + b\tan(ky))/(c + d\tan(ky))$ for some a, b, c, d, or $f(y) = (a + b\tanh(ky))/(c + d\tanh(ky))$, where $\tanh(z) = \sinh(z)/\cosh(z)$, $\sinh(z) = (\exp(z) - \exp(-z))/2)$ and $\cosh(z) = (\exp(z) + \exp(-z))/2)$.

If $f(y)$ is fractionally linear $f(y) = (a + by)/(c + dy)$ and $d \neq 0$, then the denominator is equal to zero for $y = -c/d$. The only way for the function to be defined for this y is that the numerator should also be zero, i.e., $a + by = a + b(-c/d) = 0$. But in this case $a = b(c/d)$, therefore $a + by = b(c/d + y) = (b/d)(c + dy)$, and the fraction $f(y)$ is always equal to a constant b/d. But we assumed that f is not a constant. So $d = 0$ and f is linear.

Let us prove that the expressions with tangent are impossible. Indeed, the denominator must be not identically equal to zero, therefore either $c \neq 0$, or $d \neq 0$. If $d \neq 0$, then for $ky = \arctan(-c/d)$ we have $\tan(ky) = -c/d$, and the denominator is equal to zero. As in the linear case we can then conclude that in this case f is constant, and that contradicts to our assumption that it is not. So $d = 0$ and $f(y) = (a/d) + (b/d)\tan(ky)$. Hence either $b = 0$ and $f = const$, or $b \neq 0$, and f is not defined, when $\tan(ky) = \infty$, i.e., when $ky = \pi/2$ and $y = \pi/(2k)$. So expressions with tangent are really impossible.

Let us now consider the case of hyperbolic tangent. If $k = 0$, then f is constant, which is impossible. So $k \neq 0$. If $k < 0$, then we can take $\bar{k} = -k$ and use the fact that tanh is an odd function, so $\tanh(ky) = -\tanh(\bar{k}y)$. Therefore, in the following we can assume that $k > 0$. Multiplying both the denominator and the numerator by $\cosh(z)$, we conclude that $f(y) = (a\cosh(ky) + b\sinh(ky))/(c\cosh(ky) + d\sinh(ky))$. We then substitute the expressions for sinh and cosh in terms of exp, and conclude that $f(y) = (A\exp(ky) + B\exp(-ky))/(C\exp(ky) + D\exp(-ky))$ for some A, B, C, D. Multiplying both denominator and numerator by $\exp(-ky)$, we arrive at $f(y) = (A + B\exp(-2ky))/(C + D\exp(-2ky))$. If $D = 0$, then we get a linear transformation of the exponential function. If $C = 0$, then $f(y) = (B/D) + (A/D)\exp(2ky)$, which is also a linear transformation of the exponential function. Let us now consider the case, when both C and D are different from 0.

If C and D have different signs, then for $\exp(2ky) = -D/C$ the denominator equals to zero, and so, just like in the tangent case, we conclude that f is either identically constant, or not defined in this point $y = \ln(-D/C)/(2k)$. If C and D have the same signs, then for $l = -\ln(D/C)$ we have $C + D\exp(-2ky) = C(1 + (D/C)\exp(-2ky)) = C(1 + \exp(-(2ky + l))$. If we substitute $\exp(-2ky) = \exp(-(2ky+l))\exp(l) = (C/D)\exp(-(2ky+l))$ into the numerator, we get $A+(BC/D)\exp(-(2ky+l))$, and therefore $f(y) = (A+(BC/D)\exp(-(2ky+l)))/(C(1+\exp(-(2ky + l)))$. One can check (by substituting the expression of the logistic function s_0 in terms of exp) that this expression is equal to $(A/C) + (B/D - A/C)s_0(2ky + l)$. So we get the desired expression for $K = 2k$. Q.E.D.

Proof of Theorem 2. Just like in the proof of Theorem 1, we conclude that $f(ay) = (A + Bf(y))/(C+Df(y))$. This functional equation is almost the same as the one we solved in Theorem 1, with the only exception being that here we have a product instead of a sum. It is well known that if we turn to logarithms, then the logarithm of a product is equal to the sum of logarithms. So in order to reduce this case to the one already analyzed, let us introduce the new variable $Y = \ln y$ (so that $y = \exp Y$), and a new function $F(Y) = f(\exp(y))$. For this function, the above functional equation takes the following form: for every E there exist A, B, C, D such that $F(Y + E) = (A + BF(Y))/(C + DF(Y))$ $(E = \ln a)$. In the proof of Theorem 1, we have already enumerated the solutions of this equation, so $F(Y)$ is either a fractionally-linear function, or a fractionally-linear transformation of $\tan(ky)$ or $\tanh(ky)$. If we know F, then using the equation $F(Y) = f(\exp(y))$, we can reconstruct $f(y) = F(\ln y)$ for $y > 0$. Similar expressions can be obtained for $y < 0$: in this case we need to use $Y = \ln|y|$. So in order to complete the proof, we must substitute $\ln y$ into the expressions enumerated above, and choose those that are defined everywhere.

Let us first consider the case when $F(Y) = (a + bY)/(c + dY)$. Substituting $\ln y$ instead of Y, we get $f(y) = (a + b\ln y)/(c + d\ln y)$. This function must be smooth. Let us compute the derivative f': $f'(y) = ((b/y)(c + d\ln y) - (d/y)(a + b\ln y))/(c + d\ln y)^2$. If $b/c - a/d = 0$, then, as in Theorem 1, we can conclude that f is identically constant. If $b/c - a/d \neq 0$, then for $y \to 0$ this derivative tends to ∞, so such functions f are not smooth at 0.

The fact that the expressions with tangent are impossible is proved just like in Theorem 1. So the only remaining case is the case of tanh, in which the function $F(Y)$ can be reduced to $F(Y) = (A + B\exp(-2kY))/(C + D\exp(-2kY))$. Substituting $Y = \ln y$, and using the fact that $\exp(-2k\ln y)) = (\exp(\ln y))^{-2k} = y^{-2k}$, we conclude that $f(y) = (A + By^{-\alpha})/(C + Dy^{-\alpha})$, where $\alpha = 2k$. Q.E.D.

Proof of Theorem 3. As in the proof of Theorem 1, we come to a conclusion that the optimal family F_{opt} exists and is shift-invariant. In particular, for every i the result $f_i(y + a)$ of shifting $f_i(y)$ must belong to the same family, i.e., $f_i(y+a) = C_{i1}(a)f_1(y)+C_{i2}(a)f_2(y)+\ldots+C_{im}(a)f_m(y)$ for some constants C_{ij}, depending on a. Let us first prove that these functions $C_{ij}(a)$ are differentiable. Indeed, if we take m different values y_k, $1 \le k \le m$, we get m linear equations for $C_{ij}(a)$:

$$f_i(y_k + a) = C_{i1}(a)f_1(y_k) + C_{i2}(a)f_2(y_k) + \ldots + C_{im}(a)f_m(y_k),$$

from which we can determine C_{ij} using Kramer's rule. Kramer's rule expresses every unknown as a fraction of two determinants, and these determinants polynomially depend on the coefficients. The coefficients either do not depend on a at all $(f_j(y_k))$ or depend smoothly $(f_i(y_k + a))$ because f_i are smooth functions. Therefore these polynomials are also smooth functions, and so is their fraction $C_{ij}(a)$.

We have an explicit expression for $f_i(y + a)$ in terms of $f_j(y)$ and C_{ij}. So, when $a = 0$, the derivative of $f_i(y + a)$ with respect to a equals to the derivative of the expression. If we differentiate it, we get the following formula: $f_k'(y) = c_{i1}f_1(y) + c_{i2}f_2(y) + \ldots + c_{im}f_m(y)$, where $c_{ij} = C_{ij}'(0)$. So the set of functions $f_i(y)$ satisfies the system of linear differential equations with

22-8

constant coefficients. The general solution of such system is well known [Bellman 1970], so we get the desired expressions. Q.E.D.

Proof of Theorem 4. Just like in Theorem 1, we conclude that an optimal family exists and is both shift- and scale-invariant. This means that the results of adding $\vec{a}$ to f_i and rescaling f_i by $\vec{a}$ also belong to this optimal family F.

For shift-invariance this means that

$$f_i(y_1+a_1, y_2+a_2, ..., y_p+a_p) = C_{i1}(\vec{a})f_1(y_1, y_2, ..., y_p) + ... + C_{im}(\vec{a})f_m(y_1, ..., y_p).$$

In particular, if we take $a_2 = ... = a_p = 0$, we conclude that

$$f_i(y_1+a_1, y_2, ..., y_p) = C_{i1}(a_1)f_1(y_1, y_2, ..., y_p) + C_{i2}(a_1)f_2(y_1, ..., y_p) + ... + C_{im}(a_1)f_m(y_1, ..., y_p).$$

If we fix some values $y_2, ..., y_p$ and denote $g_i(y_1) = f_i(y_1, y_2, ..., y_p)$, we conclude that $g_i(y_1+a_1) = C_{i1}(a_1)g_1(y_1) + ... + C_{im}(a_1)g_m(y_1)$. So the functions g_i satisfy the same equations that we have already solved in the proof of Theorem 3, and therefore each of g_i is equal to the linear combination of the functions $y_1^p \exp(\alpha y_1)\sin(\beta y_1 + \phi)$ from the formulation of Theorem 3.

Likewise scale-invariance means that

$$f_i(a_1y_1, a_2y_2, ..., a_py_p) = C_{i1}(\vec{a})f_1(y_1, y_2, ..., y_p) + C_{i2}(\vec{a})f_2(y_1, ..., y_p) + ... + C_{im}(\vec{a})f_m(y_1, ..., y_p).$$

If we take $a_2 = ... = a_p = 1$, we conclude that $g_i(a_1y_1) = C_{i1}(a_1)g_1(y_1) + ... + C_{im}(a_1)g_m(y_1)$. This functional equation is almost the same as for shift-invariance, the only difference is that we have a product instead of a sum. We already had such a situation, when we proved Theorem 2, and so we know what trick to apply: we must introduce a new variable $Y = \ln y_1$ (so that $y_1 = \exp(Y)$), and new functions $G_i(Y) = g_i(\exp(Y))$ (so that $g_i(y_1) = G_i(\ln y_1)$). Then for these new functions this functional equation takes the form $G_i(A+Y) = \bar{C}_{i1}(A)G_1(Y) + ... + \bar{C}_{im}(A)G_m(Y)$. This is precisely the system of functional equations that we already know how to solve. So we can conclude that $G_i(Y)$ is a linear combination of these functions $Y^p \exp(\alpha Y)\sin(\beta Y + \phi)$ from Theorem 3. When we substitute $Y = \ln y_1$, we conclude that $g_i(y_1) = G_i(Y) = G_i(\ln y_1)$ is a linear combination of the functions $(\ln y_1)^p \exp(\alpha \ln y_1)\sin(\beta \ln y_1 + \phi)$. This expression is rather complicated. The only simplification that we can apply is to change $\exp(\alpha \ln y_1)$ to $(\exp(\ln y_1))^\alpha = y_1^\alpha$, so we conclude that g_i is a linear combination of the functions $(\ln y_1)^p y_1^\alpha \sin(\beta \ln y_1 + \phi)$.

So for the same functions g_i we have two different expressions obtained from the demands of shift-invariance and scale-invariance. When can a function y_i satisfy both conclusions, i.e., belong to both classes? If it contains terms with logarithms, it cannot be a linear combination of the functions from Theorem 3, because there are no logarithms among them. The same if it contains sines of logarithms. So the only case when a linear combination of the functions $(\ln y_1)^p y_1^\alpha \sin(\beta \ln y_1 + \phi)$ is at the same time the linear combination of the functions $y_1^{\bar{\beta}} \exp(\bar{\alpha} y_1)\sin(\bar{\beta} y_1 + \bar{\phi})$ is when $p = \beta = 0$. In this case the above expression turns into y_1^α, and from the equality of these expressions we conclude that $\alpha = \bar{p}$. But $\bar{p}$ is necessarily a non-negative integer, and therefore α is non-negative integer as well. So $g_i(y_1)$, which is equal to a linear combination of such terms, is equal to the linear combination of the terms y_1^α for non-negative integers α, i.e., $g_i(y_1)$ is a polynomial. So the dependency of f_i on y_1 is polynomial.

Similarly we can conclude that f_i polynomially depends on all other variables $y_2, ..., y_p$, and therefore all the functions f_i are polynomials of y_i. Every function f from F is a linear combination of these polynomials, and therefore a polynomial itself. Q.E.D.

Proof of Theorem 5. We can repeat the proof of Theorem 1 and come to a conclusion that all these functions are either linear, or exponential, or a logistic function. Neither linear, nor exponential functions are bounded, so only a logistic function is left. Q.E.D.

22-9

Acknowledgments. This research was supported by NSF grant No. CDA-9015006, NASA Research grant NAG 9-482 and a grant from the Institute for Manufacturing and Materials Management. The authors are also thankful to Dr. R. Hecht-Nielsen (San Diego, CA) for encouragement, to Dr. G. E. Hinton (Toronto) for interesting preprints, and to all participants of the NSF II Workshop (Purdue, 1991), especially to Dr. Lokendra Shastri (Philadelphia) for valuable discussions.

References

Aczel, J. *Lectures on functional equations and their applications.* Academic Press, NY-London, 1966.

Bellman, R. *Introduction to matrix analysis.* McGraw-Hill, N. Y., 1970.

Braham, R. *Numerical analysis of neural networks*, in Proceedings of the International Joint Conference on Neural Networks, IEEE Press, Washington, DC, 1989, pp. 1428–1432.

Braham, R. and J. O. Hamblen *The design of a neural network with a biologically motivated architecture.* IEEE Transactions on Neural Networks, 1990, Vol. 1, No. 3, pp. 251–262.

Chevalley, C. *Theory of Lie groups*, Princeton University Press, Princeton, NJ, 1946.

Cowan, J. D. *A mathematical theory of central nervous activity.* Ph. D. Dissertation, Univ. London, 1967.

Guillemin, V. M. and S. Sternberg. *An algebraic model of transitive differential geometry*, Bulletin of American Mathematical Society, 1964, Vol. 70, No. 1, pp. 16–47.

Grossberg, S. *Nonlinear neural networks: Principles, mechanisms and architectures.* Neural Networks, 1988, Vol. 1, pp. 17–61.

Hecht-Nielsen, R. *Kolmogorov's mapping neural network existence theorem.* Proceedings of IEEE International Conference on Neural Networks, 1987, vol. 3, pp. 11–13.

Hecht-Nielsen, R. *Neurocomputing.* Addison-Wesley, Reading (MA), 1991.

Kosko, B. *Neural networks and fuzzy systems.* Prentice Hall, Englewood Cliffs, NJ, 1992.

Kreinovich, V. A mathematical supplement to the paper: I. N. Krotkov, V. Kreinovich and V. D. Mazin. *A general formula for the measurement transformations, allowing the numerical methods of analyzing the measuring and computational systems.* Measurement Techniques, 1987, No. 10, pp. 8–10.

Kreinovich, V. *Group-theoretic approach to intractable problems.* Lecture Notes in Computer Science, Springer-Verlag, Berlin, Vol. 417, 1990, pp. 112–121.

Kreinovich, V. *Arbitrary nonlinearity is sufficient to present all functions by neural networks: a theorem.* Neural Networks, 1991, vol. 4, pp. 381–383.

Kreinovich, V. and J. Corbin, *Dynamic tuning of communication network parameters: why fractionally linear formulas work well.* University of Texas at El Paso, Computer Science Department, Technical Report UTEP-CS-91-4, 1991.

Kreinovich, V. and S. Kumar, *Optimal choice of &- and ∨- operations for expert values.* Proceedings of the 3rd University of New Brunswick Artificial Intelligence Workshop, Fredericton, New Brunswick, Canada, 1990, pp. 169–178.

Kreinovich, V. and S. Kumar, *How to help intelligent systems with different uncertainty representations communicate with each other.* Cybernetics and Systems: International Journal, 1991, vol. 22, No. 2, pp. 217–222.

Kreinovich, V. and C. Quintana. *How does new evidence change our estimates of probabilities: Carnap's formula revisited.* Submitted to Cybernetics and Systems.

22-10

Minsky, M. and S. Papert. *Perceptrons*. MIT Press, Cambridge (MA), 1968.

Munro, P. W. *State-dependent factors influencing neural plasticity: a partial account of the critical period*. In: J. L. McClelland, D. E. Rumelhart et al *Parallel distributed processing*, MIT Press, Cambridge, MA, 1986, Vol. 2, pp. 471–487.

Singer, I.M. and S. Sternberg. *Infinite groups of Lie and Cartan*, Part 1, Journal d'Analyse Mathematique, 1965, Vol. XV, pp. 1–113.

Wasserman, P. *Neural computing: Theory and Practice*. Van Nostrand Reinhold, N.Y., 1989.

Wiener, N. *Cybernetics, or Control and Communication in the animal and the machine*, MIT Press, Cambridge MA, 1962.

(22-11)